

Claims

What is claimed is:

- [c1] A method for transparently optimizing data access, comprising:
gathering information related to data usage when a system is processing; and
determining a usage pattern of the system using gathered information.
- [c2] The method of claim 1, further comprising:
pre-fetching data determined by the usage pattern of the system;
caching data locally;
accessing data by the system; and
synchronizing cached data with persistent data.
- [c3] The method of claim 1, further comprising:
generating a description of a business process model by retaining the usage pattern over a
period of at least one execution of the system.
- [c4] The method of claim 1, further comprising:
generating tests for the system by retaining the usage pattern over a period of at least one
execution of the system.
- [c5] The method of claim 1, wherein the usage pattern comprises pieces of information used
together based on a relationship.
- [c6] The method of claim 5, wherein the relationship is temporal.
- [c7] The method of claim 5, wherein the relationship is causal.
- [c8] The method of claim 1, further comprising:
deriving an initial usage pattern from application code analysis.
- [c9] The method of claim 1, further comprising:
deriving an initial usage pattern from an empty set.

- [c10] The method of claim 1, further comprising:
deriving an initial usage pattern from a specification.
- [c11] The method of claim 1, further comprising:
displaying the usage pattern to a display device.
- [c12] The method of claim 1, further comprising:
generating documentation from the usage pattern.
- [c13] A method for transparently optimizing data access, comprising:
gathering information related to data usage when a system is processing;
determining a usage pattern of the system using gathered information;
pre-fetching data determined by the usage pattern of the system;
caching data locally;
accessing data by the system; and
synchronizing cached data with persistent data.
- [c14] A method for transparently optimizing a distributed application having a client-side and a server-side, comprising:
gathering information related to data usage on the client-side when the distributed application is processing; and
determining a usage pattern using gathered information.
- [c15] The method of claim 14, further comprising:
pre-fetching data from the server-side;
caching data on the client-side;
accessing data on the client-side; and
synchronizing cached data on the client-side with persistent data on the server-side.
- [c16] The method of claim 14, wherein the usage pattern comprises pieces of information used together based on a relationship.
- [c17] The method of claim 16, wherein the relationship is temporal.

- [c18] The method of claim 16, wherein the relationship is causal.
- [c19] The method of claim 14, wherein the data represents objects.
- [c20] The method of claim 14, further comprising:
deriving an initial usage pattern from application code analysis.
- [c21] The method of claim 14, further comprising:
deriving an initial usage pattern from an empty set.
- [c22] The method of claim 14, further comprising:
deriving an initial usage pattern from a specification.
- [c23] The method of claim 14, further comprising:
displaying the usage pattern to a display device.
- [c24] The method of claim 14, further comprising:
generating documentation from the usage pattern.
- [c25] A method for transparently optimizing a distributed application having a client-side and a server-side, comprising:
gathering information related to data usage on the client-side when the distributed application is processing;
determining a usage pattern using gathered information;
pre-fetching data from the server-side;
caching data on the client-side;
accessing data on the client-side; and
synchronizing cached data on the client-side with persistent data on the server-side.
- [c26] A computer-readable medium having recorded thereon instructions executable by processing, the instructions for:
gathering information related to data usage when a system is processing;
determining a usage pattern of the system using gathered information;
pre-fetching data determined by the usage pattern of the system;

caching data locally;
accessing data by the system; and
synchronizing cached data with persistent data.

- [c27] An apparatus for transparently optimizing data access, comprising:
- means for gathering information related to data usage when a system is processing;
 - means for determining a usage pattern of the system using gathered information;
 - means for pre-fetching data determined by the usage pattern of the system;
 - means for caching data locally;
 - means for accessing data by the system; and
 - means for synchronizing cached data with persistent data.